LA-UR-02-3163

Title: Reliability of Programmable Input/Output Pins in the Presence of Configuration Upsets

Author(s): Michael Wirthlin, BYU
Nathan Rollins, BYU
Michael Caffrey, 111180, NIS3
Paul Graham, 181856, NIS3
Los Alamos National Laboratory

Los Alamos
NATIONAL LABORATORY

Reliability of Programmable Input/Output Pins in the Presence of Configuration Upsets

Nathan Rollins[1],Michael J. Wirthlin[1], Michael Caffrey[2],
and Paul Graham[2]

nhr2@ee.byu.edu, wirthlin@ee.byu.edu, mpc@lanl.gov, and *grahamp@lanl.gov*

[1]Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT

[2]Los Alamos National Laboratory, Los Alamos, NM

Abstract

Field programmable gate arrays (FPGAs) are an attractive alternative for space-based remote sensing applications. However, SRAM-based FPGAs are sensitive to radiation-induced single-event upsets within the configuration memory. Such configuration upsets may change the logic, routing, and operating modes of a user FPGA design. Upsets within the configuration of an I/O block are especially troublesome as they may impact the operation of other system components. This paper will evaluate the operation of the I/O block within the Xilinx Virtex FPGA in the presence of configuration memory upsets and introduce techniques for detecting and repairing such failures.

1 Introduction

Field programmable gate arrays (FPGAs) are an attractive alternative for application-specific processing in space based applications because of their flexibility and in-system re-programmability. Unlike application-specific integrated circuits (ASICs), FPGAs can be configured and reprogrammed while the spacecraft is in orbit. Multiple application-specific circuits can be configured within the FPGA based on changing mission needs or improvements to the algorithm. Several projects report the advantages of reconfigurable FPGAs within a spacecraft [1, 2, 3].

While FPGAs offer several advantages for space-based computing, FPGAs are sensitive to radiation effects. Reconfigurable SRAM based FPGAs are especially sensitive to single-event upsets or SEUs. A single-event upset is a change of state within a circuit caused by a heavy ion or proton. Single-event upsets cause a temporary, non-permanent state change within the flip-flops, latches, and memories of a device. While single-event upsets do not permanently damage a device, the change in state caused by an SEU may change the operation of the device or lead to undesired system behavior.

Single-event upsets are especially troublesome when they occur in the configuration memory of the FPGA. The configuration memory of an FPGA defines the routing, logic function, and operating modes of the circuit configured onto the device. Upsets in the configuration memory may *change* the circuit operating on the device. Such configuration SEUs may disconnect circuit routing, change internal logic equations, or modify the operating modes of memories and other FPGA resources. To operate properly in space, FPGA-based systems must anticipate and mitigate against configuration memory SEUs.

While all FPGA resources are sensitive to configuration SEUs, it is especially important to insure that the programmable input/output blocks (IOBs) operate correctly in the presence of configuration SEUs. A potentially

dangerous situation can result if a configuration memory upset changes the function a programmable IOB. For example, modifications in the configuration memory of an IOB may change the programmable IOB pin from an input to an output. The creation of an unintended output IOB may cause unanticipated contention on a system bus, excessive power consumption, and incorrect system operation.

This paper investigates the behavior of programmable IOBs in the presence of configuration memory upsets. Specifically, this paper will study the behavior of IOBs found within the Xilinx Virtex FPGA. To study the IOB behavior, a simulation environment was created to artificially corrupt the IOB configuration memory. The paper will begin by reviewing related work and introducing the SEU simulator. Next, the methodology for testing programmable IOBs will be described along with test results. After presenting the results, techniques for detecting faults within the programmable IOBs will be suggested.

2   Effects of SEUs in FPGAs

With increased interest in exploiting programmable logic in space related applications, researchers have investigated the suitability of commercially available FPGAs in radiation environments[1]. Interest in the Xilinx Virtex FPGA architecture has motivated the testing of the high-reliability XQVR Virtex device for radiation tolerance [4]. Based on a thin-epitaxial 0.22 μmm CMOS process, this device tolerates a total dose in the range of 80 to 100 krads(Si). This total dose tolerance is acceptable for many low-earth orbit applications[5]. Further, there is no risk of latch-up occurring in orbit.

While the XQVR Virtex FPGA is immune to latch-up and has an acceptable total-dose tolerance, it is sensitive to heavy ion and proton induced SEUs[6]. A single-event upset may alter the state of the FPGA flip-flops or configuration memory causing the device to pro-duce incorrect results or unspecified behavior. Because of the high-density of configuration memory, SEUs are especially troublesome for SRAM-based FPGAs. The internal configuration memory of the Virtex V1000 FPGA device requires almost 6 million bits to describe the interconnect, logic, I/O pins, and operating modes of a user design [7].

Several techniques have been proposed and tested for mitigating SEUs in FPGAs. Several techniques use hardware redundancy to reduce the probability of failure [8]. By replicating the desired circuitry and comparing the results, faults in the configuration can be detected and reported. Other techniques rely on device reconfiguration to continually "clean" the configuration bitstream[9]. By repeatedly configuring the device, SEUs occurring within the configuration bitstream are replaced by the correct value.

3   Xilinx Virtex IOB Architecture

It is especially important to consider how an upset in the configuration memory can alter the operation of the input/output pins of an FPGA. Since FPGA pins are attached to other devices within a system, a change in the IOB behavior may have a large impact on the electronic system. For example, an IOB pin configured as an input may be changed into an output pin through unintentional configuration memory SEUs. The creation of such unintentional output pins may produce contention on a system bus and generate excessive current within the FPGA.

The Xilinx Virtex IOB is complex and supports many modes of operation and customization options. Figure 1 shows a simplified model of the IOB architecture. Each IOB can be configured as an input, an output, or as a bi-directional tri-state signal. When the IOB is configured as an input, a de-asserted tri-state enable signal disables the tri-state buffer (see

Figure 2). When the IOB is configured as an output, the tri-state signal is asserted low to enable the tri-state buffer. In addition, each IOB includes three flip-flops to provide registering capability at the input, output, and tri-state enable.
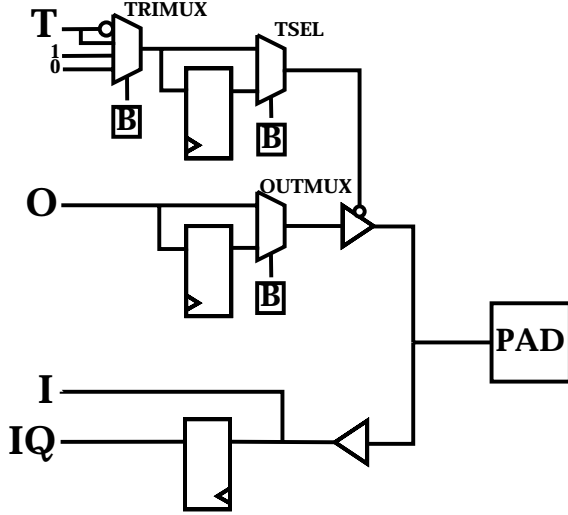

Figure 1: Simplified Programmable I/O Block

For clarity, many IOB circuit elements have been left out of Figure 1, however, there are many different additional elements included in each Virtex IOB. For instance, the IOB's I/O standard can be changed from LVTTL (the default I/O standard) to one of 15 other standards. Another programmable IOB element is the output signal drive strength. This drive strength can range from 2mA to 24mA. Also included in the complexity of each IOB is an optional termination (pullup resistor, pulldown resistor or keeper), an optional input delay and a variable slew rate.

In the Xilinx Virtex XCV1000 device there are approximately 324 bits in the FPGA configuration bitstream which correspond to each IOB circuit element. Upsetting any of these 324 bits can have adverse effects on the behaviour of the IOB[1]. Upsets in the IOB bitstream can affect the IOB configuration modes, I/O logic values, or any of the architectural elements and attributes of the IOB. For example, output logic errors can be caused if the output

polarity is inverted. Also, a change in the slew rate or clock polarity can cause timing errors. Upsetting the IOB bitstream can have more severe consequences if changes could cause contention on the system bus. This can happen if the tri-state buffer of an input IOB is enabled. In this scenario, an input IOB becomes corrupted and acts as an output IOB.
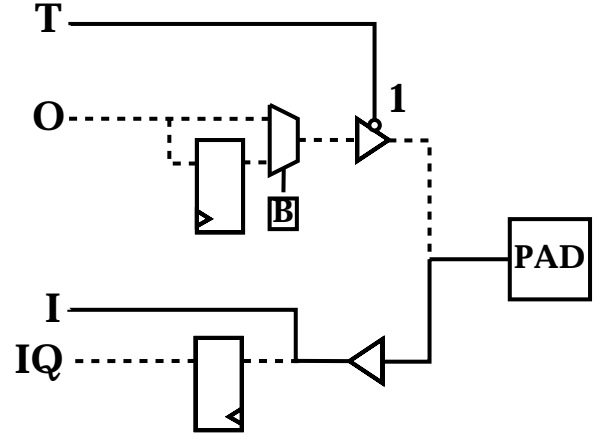

Figure 2: Simplified Input IOB

It is important to note that many of the IOB architectural attributes correspond to a single bit in the configuration bitstream. Thus an SEU in the IOB section of the configuration memory can very easily affect how an IOB functions. For example, if the bit(s) reserved for the tri-state multiplexer (TRIMUX) select line are corrupted and the signal is inverted, the operation of the tri-state buffer will change. This could potentially cause bus contention. To illustrate, consider Figure 2 which shows an IOB functioning as an input. Figure 3 shows how IOB configuration upsets can activate the tri-state buffer causing the output signal to be driven onto the bus, in contention with the incoming input signal.

4 IOB Corruption Tests

The primary purpose of this work is to determine the robustness of programmable IOBs in the presence of upsets within the configuration memory. To test the reliability of the Xilinx
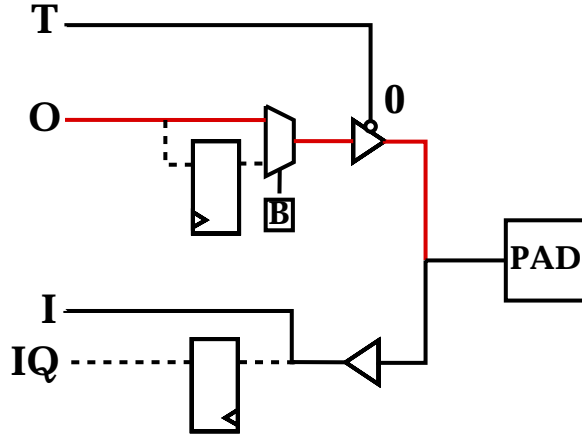
Figure 3: Corrupted input IOB

Virtex IOB architecture, upsets will be artificially inserted in the appropriate IOB configuration memory bits. As the IOB configuration memory is corrupted, the operation of the IOB is carefully monitored to detect changes in the IOB behavior. Specifically, these tests will determine how many configuration bit upsets are required to reconfigure an input IOB (i.e. Figure 2) into an output IOB as shown in Figure 3.

This IOB testing methodology will use a simulator developed specifically for testing upsets within the FPGA configuration memory [10]. Based on the SLAAC-1V FPGA computing board[11], this test-bed provides the capability of rapidly inserting or removing upsets within the configuration memory of a target FPGA. The SLAAC-1V board contains two Xilinx Virtex V1000 FPGAs for testing purposes (PE1 and PE2) and one FPGA (PE0) for observing the FPGAs under test and providing appropriate circuit stimulus. This simulation platform also provides the ability to control and query the operation of the FPGAs from a host-based computer.

The primary goal of this test is to determine how many configuration bit upsets are required to change an input IOB into an output IOB. To perform this test, a simple FPGA design is created that contains only two input IOBs as shown in Figure 4. This design is config-

ured onto PE1 and will be subject to a series of artificial configuration memory upsets. As configuration upsets are inserted into these two input IOBs, the behavior of the these IOBs will change.
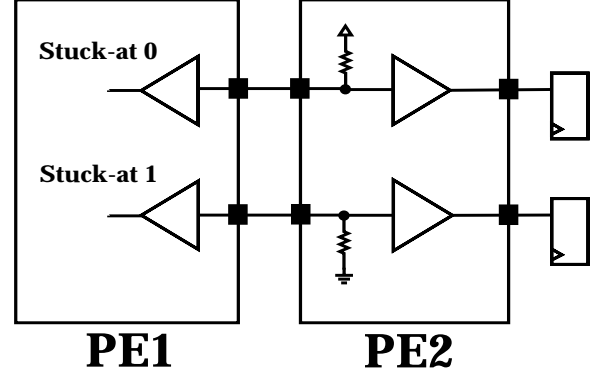

Figure 4: Programmable I/O Test Architecture

To detect changes in the operation of the IOBs within PE1, a second FPGA design is created in PE2. As shown in Figure 4, this design contains two input IOBs that are connected to the two input IOBs of PE1 through traces on the board. These input pads monitor the signal values of PE1 and are able to detect changes in the input/output behavior of the IOBs under test. These two signals are also readable by the host computer. This allows the host to detect changes in the input/output behavior of the design under test.

Under normal circumstances, no active circuit is driving the signal trace connecting the input pads in PE1 and PE2. To force a default value on each of these wires, internal resistors are added. A pull-up resistor ($\approx 13k\Omega$) is added to the first wire forcing a logic '1' and a pull-down resistor ($\approx 22k\Omega$) is added to the second wire forcing a logic '0'. When the IOBs in PE1 are operating correctly, the host will read a logic '1' for the first pair of I/O blocks and a logic '0' for the second pair of I/O blocks.

The test structure of Figure 4 is organized to detect changes in the input IOB configuration of PE1. Specifically, this test detects when configuration upsets change the IOB from an

input into an output. If the input IOBs in PE1 are changed from an input to an active output, the output IOB will overcome the pull-up or pull-down and drive a new value on the signal input to PE2. This change in signal level is detected by the host and recorded.

Two signal levels are used in this test structure to identify both the stuck-at-0 and stuck-at-1 cases. If the IOB is changed to an output driving a logic '1' (i.e. stuck-at-1), it will only be detected by the input pad with the pull-down. Since the default value of the pull-down signal is '0', a stuck-at-1 output driver will overcome the pull-down and drive a logic '1'. This change in value is detected by the host. However, this fault will not be detected by the input pad with the pull-up resistor. A stuck-at-1 output driver will continue to drive a logic '1' on the pull-up signal with no change in value. By using two default signal levels, both output driver faults can be detected.

5   IOB Test Results

With the FPGAs configured as shown in Figure 4, configuration bits associated with PE1 are modified in an attempt to convert the input IOBs into output IOBs. Two series of tests are performed on the IOBs. In the first series of tests, each of the 324 configuration bits associated with the two IOBs are individually toggled. The purpose of this test is to determine if any single-bit upset within the configuration bitstream will change the input IOB structure into an output IOB structure. In the second series of tests, all two-bit combinations of the 324 configuration bits are toggled. This test will determine how many two-bit configuration upsets change the input IOB into an output. While additional tests could be performed to investigate triple and quadruple configuration bit upsets, this work is limited to the study of single and double upsets within the configuration bitstream.

5.1   Single-Bit Results

In the first series of tests, all single bit upsets in the configuration bitstream are made for the two IOBs under test in PE1. For each test, a single bit in the bitstream is toggled and configured onto the FPGA. Once the modified design is operating within PE1, the values of the two test signals are queried by the host. After recording the result of the test, the corresponding bit within in the bitstream is repaired and the FPGA is configured with the original design. This process is repeated until all 324 bits associated with each of the two IOBs have been tested.

After performing this test for each of the 324 configuration bits, only one bit changed the default values of the two test signals. When this particular configuration bit was set, a logic '1' was read from the default logic '0' case. This configuration bit changed in the input IOB of PE1 in a way that overcomes the pull-down in PE2. Upon further investigation, it is determined that this particular configuration bit enables the pull-up resistor within the IOB of PE1 as shown in Figure 5. Since the pull-up is slightly stronger than the pull-down (i.e. $13k\Omega$ vs. $22k\Omega$), a logic '1' is read in PE2.
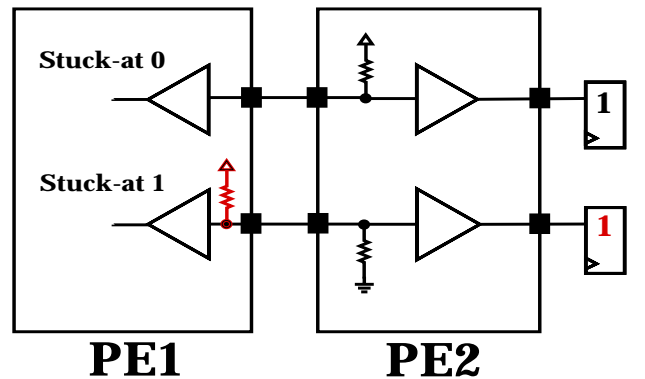
Figure 5: Single-Bit SEU Activating a Pull-Up Resistor.

While this particular fault within the IOB does cause a logic error and possibly an incorrect voltage level, it does not create a situation resulting in excessive IOB current. The

current is limited below .1 mA by the series resistance of the relatively weak pull-up and pull-down IOB resistors. This particular fault will not cause damage or excessive current to either the FPGA or other non-FPGA components connected to this pin.

It is interesting to note that no single-bit configuration upset changes the value of the default logic '1' circuit. While it is likely that a bit within the configuration bitstream enables the pull-down resistor within the IOB, the pull-down resistor is weaker than the pull-up and the fault is not detected. Again, this situation results in minimal current and will not damage the device or external circuitry.

5.2 Two-Bit Results

In the second series of tests, all two-bit pairs of the IOB configuration bitstream are modified and programmed onto the test circuit of Figure 4. For each of these tests, two configuration bits are toggled and the bitstream is configured onto the device. Once the circuit is operational, the values of the default test signals are read by the host and recorded. This process is repeated until all 324×323 or 104,652 pairs associated with each of the two IOBs have been tested.

As expected, corrupting two bits within the IOB configuration bitstream identified more IOB faults than the single-bit test. Most of these observed faults occurred when the pull-up associated with the default Logic '0' test signal was enabled (see Figure 5). Since this particular fault was already identified in the single-bit case, double-bit variations of this fault are ignored.

The two-bit configuration upset tests identified a fault in the IOB that was not seen in the single-bit test. In this instance, the input IOB in PE1 associated with the default '1' signal was changed to an output IOB *actively* driving a logic '0' on the test wire. As shown in Figure 6, this active driver overcomes the passive pull-up resistor to force a '0' on the signal wire.
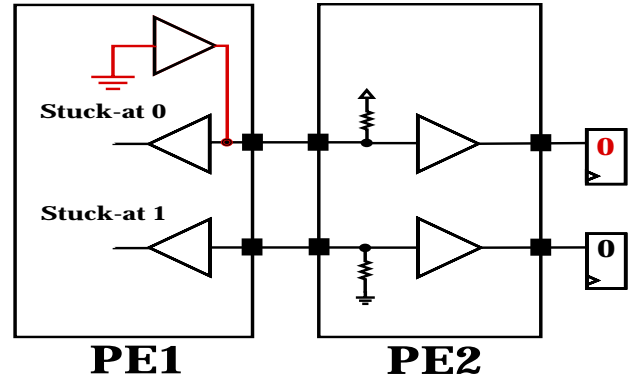

Figure 6: Double-Bit SEU Activating an Output Logic '0' Driver.

The unintended active output driver shown in Figure 6 occurs for two pairs of configuration upsets. It is interesting to note that one of the modified IOB configuration bits is common to both cases. While the actual purpose of these configuration bits is not published, it is likely that these bits affect the TRIMUX select line in Figure 1 and set the drive strength of the output driver.

Based on these results, it is clear that the IOB is not immune to two-bit upsets within the configuration memory. Two of the 104,652 pairs of double-bit upsets will cause the IOB to change from an input to an active output driver. If configuration memory upsets are distributed uniformly over the entire configuration bitstream, the probability of upsetting these two specific configuration bits is very low. With 5,962,944 bits within the complete Xilinx Virtex V1000 FPGA, the probability that two configuration upsets will cause this IOB failure within one of the 512 IOBs is 2.9×10^{-11}.

6 Detection of IOB Configuration Upsets

Although the probability is low that a two-bit configuration upset will cause the creation of an unintended output, it is important that such a condition is detected by a radiation tolerant system. If such unintended outputs

are not detected and corrected, the electronic system may suffer incorrect system operation, premature damage, and excessive power consumption. Once a fault within the IOB has been detected, the FPGA configuration memory must be repaired through device reconfiguration. Two techniques for detecting upsets within the IOB configuration will be presented.

6.1 Redundant I/O Pins

The first technique for detecting configuration upsets within the IOB is the use of redundant I/O pins. The circuit shown in Figure 7 demonstrates how two FPGA pins can be used to detect changes in the behavior of the IOBs. In this circuit, a signal is driven by an external device and connected to two independent FPGA input pads through two separate current-limiting resistors.

Within the FPGA, these two internal signals are constantly compared using available logic resources. When functioning normally, the two input pads will read the same value from the two external resistors. The XOR comparator is used to compare the two internal signals – as long as the two input signals are the same, the XOR produces a logic '0'. The output of the XOR is registered by the system clock to avoid false errors caused by timing differences between the two input signals. A logic '0' within this register indicates correct operation of the I/O pads.
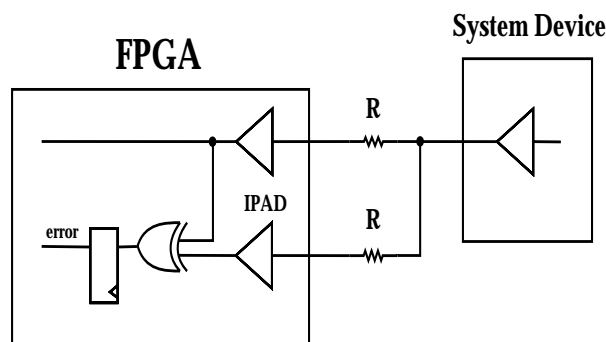

Figure 7: Redundant Input Pads

The circuit is designed to detect logic differences between the two I/O pins caused by a

change in the IOB configuration. If the configuration of one of these input IOBs is changed into an active output pad, the output pad will drive an active signal on the FPGA-side of the resistor. If the external device and the FPGA pin are driving different logic values, current will flow through the resistor and an incorrect logic level will be driven by the FPGA output pad. This incorrect value will be read by the pad input and the internal comparator circuit will detect a difference between the corrupted IOB and the uncorrupted IOB. Once the error has been registered and detected, the host will completely reconfigure the device to repair the configuration memory upsets.

The primary advantage of this technique is the speed at which the IOB fault is detected. This circuit will detected the IOB fault as soon as the comparator circuit detects differences in the two input signal values. Rapid detection of the fault will prevent excessive power loss and limit the damage caused by the contention. An additional advantage of this approach is the current limitations placed on the unintended FPGA output. If an FPGA output pad contends with a pin from an external device, the current during contention is limited by the series resistor.

The primary disadvantage of this technique is the added hardware cost. To implement this technique, two FPGA pins are required for each FPGA input signal. Many FPGA designs are limited by the available I/O resources and cannot tolerate redundant I/O signals. Other disadvantages of this approach include increased signal transition time and static power consumption caused by the series signal resistors.

6.2 Configuration Readback

Another technique for detecting upsets within the IOB configuration memory is the use of configuration readback. The Xilinx Virtex family of FPGAs supports the ability to read the current contents of the configuration memory through a process call "readback". This

capability allows the user of the FPGA to determine if the correct configuration bitstream was loaded into the FPGA. Readback has also been used to detect radiation-induced upsets within the configuration memory [9].

To detect random upsets within the configuration memory of the FPGA, the contents of the configuration memory must be constantly read through the readback process. As the bitstreams are read from the device, they must be compared with known "safe" configuration bitstreams. If the bitstream read from the device differs from the known "safe" bitstream, an upset within the configuration is detected and the device can be reconfigured.

One disadvantage of this approach is the relatively long time required to detect and identify configuration upsets. The detection process must read each frame of the bitstream, compute a cyclic-redundancy code (CRC), and compare it with a known CRC code-book. This process can take several hundred milliseconds for the entire device.

An important advantage of this technique is that readback will detect upsets within the *entire* configuration memory, not just the input/output blocks. Constant readback and verification of the bitstream can be employed for SEU detection for the entire FPGA and avoid the need to implement architectural specific SEU detection. Configuration faults within the interconnect, logic, flip-flops, clock buffers and other global FPGA resources can all be detected using the same readback-compare methodology.

7   Conclusion

The test methodology described in this paper has been used to determine the behavior of programmable input/output blocks in the presence of configuration memory upsets. Specifically, this test procedure was used to determine that at least two bits within the configuration memory must be upset before an input pin is changed into an active output pin. In this case, the unintentional active output drives a constant '0' signal (stuck at 0) and will likely cause contention on a system bus.

Two techniques were presented for detecting such faults within the IOB configuration. Redundant I/O pins and configuration readback can be used to identify faulty IOBs and signal a need for device reconfiguration. Once a fault has been found within an IOB, the faulty IOB must be repaired by reloading the configuration and resetting the device. Rapid repair of such faults is necessary to minimize the power consumption, premature device failure, and improper system operation caused by a faulty IOB.

References

[1] R. Katz, K. LaBel, J.J. Wang, B. Cronquist, R. Koga, S. Penzin, and G. Swift. Radiation effects on current field programmable technologies. *IEEE Transactions on Nuclear Science*, 44(6):1945–1956, December 1997.

[2] J. Wang, R. Katz, J. Sun, B. Cronquist, J. McCollum, T. Speers, and W. Plants. SRAM based re-programmable FPGA for space applications. *IEEE Transactions on Nuclear Science*, 46(6):1728–1735, December 1999.

[3] Michael Caffrey. A space-based reconfigurable radio. In Toomas P. Plaks and Peter M. Athanas, editors, *Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA)*, pages 49–53. CSREA Press, June 2002.

[4] E. Fuller, M. Caffrey, A. Salazar, C. Carmichael, and J. Fabula. Radiation testing update, seu mitigation, and availability analysis of the virtex fpga for space reconfigurable computing. In *4th Annual Conference on Military and Aerospace*

Programmable Logic Devices (MAPLD), page P30, 2000.

[5] Phil Brinkley and Carl Carmichael. SEU mitigation design techniques for the XQR4000XL. Technical report, Xilinx Corporation, March 15, 2000. XAPP181 (v1.0).

[6] E. Fuller, M. Caffrey, P. Blain, C. Carmichael, N. Khalsa, and A. Salazar. Radiation test results of the Virtex FPGA and ZBT SRAM for space based reconfigurable computing. In *MAPLD Proceedings*, September 1999.

[7] Xilinx, Inc. *Virtex Series Configuration Architecture User Guide, Xilinx Application Notes 151, v1.5*, September 2000.

[8] Carl Carmichael. Triple module redundancy design techniques for virtex FP-GAs. Technical report, Xilinx Corporation, November 1, 2001. XAPP197 (v1.0).

[9] Carl Carmichael, Michael Caffrey, and Anthony Salazar. Correcting single-event upsets through virtex partial configuration. Technical report, Xilinx Corporation, June 1, 2000. XAPP216 (v1.0).

[10] Eric Johnson, Michael J. Wirthlin, and Michael Caffrey. Single-event upset simulation on an FPGA. In Toomas P. Plaks and Peter M. Athanas, editors, *Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA)*, pages 68–73. CSREA Press, June 2002.

[11] USC-ISI East. *SLAAC-1V User VHDL Guide*, October 1, 2000. Release 0.3.1.